

PX9

**EXPERT REPORT OF MAURICE P. HERLIHY IN
SECURITIES AND EXCHANGE COMMISSION V.
TELEGRAM GROUP INC. AND TON ISSUER INC.,
19-CV-9439 (PKC)**

December 27, 2019

TABLE OF CONTENTS

Contents

1.	INTRODUCTION	1
1.1.	Assignment	1
1.2.	Qualifications	1
1.3.	Documents Relied Upon	2
2.	PROCESS OF ANALYSIS	2
3.	CASE BACKGROUND	3
4.	SUMMARY OF FINDINGS	3
5.	OVERVIEW OF BLOCKCHAINS AND CRYPTOCURRENCIES	5
6.	TECHNICAL OVERVIEW OF TON BLOCKCHAIN	8
a.	Types of Nodes	8
b.	Scalability Mechanisms	10
c.	Communication Protocols.....	11
7.	EVALUATION OF TON BLOCKCHAIN CODE	11
8.	CONCLUSIONS	13

1. INTRODUCTION

1.1. ASSIGNMENT

1. I have been engaged by the Securities and Exchange Commission (“SEC”) to provide expert testimony in the matter of *Securities and Exchange Commission v. Telegram Group Inc. and TON Issuer Inc.*, 19-cv-9439 (PKC).

1.2. QUALIFICATIONS

2. I hold the *An Wang Chair* of Computer Science at Brown University. I have an A.B. in Mathematics from Harvard University, and a Ph.D. in Computer Science from M.I.T. I have served on the faculty of Carnegie Mellon University and the staff of DEC Cambridge Research Lab. I am the recipient of the 2003 Dijkstra Prize in Distributed Computing, the 2004 Gödel Prize in theoretical computer science, the 2008 International Symposium on Computer Architecture influential paper award, the 2012 Edsger W. Dijkstra Prize, and the 2013 Wallace McDowell award. I received a 2012 Fulbright Distinguished Chair in the Natural Sciences and Engineering Lecturing Fellowship, and I am a fellow of the Association for Computing Machinery, a fellow of the National Academy of Inventors, the National Academy of Engineering, and the National Academy of Arts and Sciences. I have served as a consultant for Facebook’s Libra coin and Algorand’s Algo coin

3. My curriculum vitae, attached as Appendix A to this report, provides more details about my educational and professional background and experience, as well as a summary of my publications. I have been retained through Integra FEC, a forensic data analytics and litigation consulting firm. Integra will be compensated at the rate of \$400 per hour for my work and I will be compensated by Integra at the rate of \$350 per hour. A data analyst and a Vice-President at Integra also performed work in connection with this report. Integra will be compensated by the

SEC for their work at the rate of \$220 per hour for data analyst and \$420 per hour for the Vice-President.

1.3. DOCUMENTS RELIED UPON

4. I relied on the SEC's Complaint in this case, copies of Telegram Group Inc. and TON Issuer Inc's ("Defendants") Purchase Agreements, source code downloaded from <https://github.com/ton-blockchain/ton> ("public release"), which purports to be the current but not necessarily final computer code for a planned "Telegram Open Network ('TON') Blockchain," all the documents found on <https://test.ton.org/> ("public documents") and the Defendants' Interrogatory responses in this action.

2. PROCESS OF ANALYSIS

5. The public release consists of about 200,000 lines of code. An initial inspection showed that much of this code performs routine and mundane functions such as formatting data for message transmission, interacting with third-party databases, storing cryptographic libraries, and so on. These functions would be performed in essentially the same way in any blockchain. I focused on locating the code for the functions that distinguish the TON Blockchain from its competitors, and whose secure and efficient design and implementation would be essential to the blockchain's success. Working with an Integra Data Analyst, I installed and ran a "lite node" and the Integra Data Analyst ran a "full node" that participated in the current test version of the TON Blockchain¹. Our goal was to estimate the eventual performance and scalability of a full running TON Blockchain, but we were unable to make such an estimate because the current testnet release is so much smaller in scale (fewer nodes and transactions) than the anticipated full TON

¹ A lite node tracks minimal information about recent activity only, while a full node tracks complete information. A lite node could run on a laptop, while a full node requires more heavyweight hardware resources.

Blockchain. For example, only 36 validator nodes were detected, while the public documents project as many as 1000 validators when the system is fully deployed.

6. Working with an Integra Data Analyst, we also examined the web pages of certain TON Blockchain application providers identified in the Defendants' Interrogatory responses.

3. CASE BACKGROUND

7. The SEC has filed a Complaint alleging that the Defendants violated Section 5 of the Securities Act of 1933 by engaging in unlawful offers and sales of digital tokens known as Grams. The Complaint alleges that the Defendants used part of the proceeds from their sale of Grams to the "Initial Purchasers" to capitalize their business and finance the creation of the Telegram Open Network blockchain (the "TON Blockchain"). The Complaint also alleges that the Defendants promised to deliver Grams to the Initial Purchasers in conjunction with the network launch of the TON Blockchain. The Purchase Agreements between the Defendants and the Initial Purchasers defined the "Network Launch" to mean "the public release of the version of the TON Network after completion of the test launch and security audits, as determined by the Issuer in its sole discretion." The Complaint also alleges that the Defendants circulated certain promotional materials, including versions of a technical "White Paper," "Teasers, and "Primers," describing the efforts the Defendants would engage in to develop the TON Blockchain, including various applications and features it expected to run on it, and to integrate the TON Blockchain with Telegram Messenger, a mobile messaging application owned by Telegram, via a wallet application for the TON Blockchain.

4. SUMMARY OF FINDINGS

8. Based on my i) inspection of the source code in the public release, ii) the public documents, iii) the Defendants' Interrogatory responses, iv) my academic research in this field,

and v) technical assistance from Integra as detailed below, I have reached the following conclusions:

- The publicly-released “testnet” version of the TON Blockchain code, while complete enough to run simple transactions on simulated assets, lacks critical components that would be required in a fully developed and running system, such as i) the core “BFT Consensus” protocol, ii) the code that controls validator selection, iii) code that controls validator incentives and rewards, iv) the code that controls how validator misbehavior is detected and deterred, and v) the “vertical blockchain” mechanism for repairing corrupted blocks. Moreover, there is no systematic analysis proving that either the design or specific implementation of these components is secure and correct. I, and in my opinion potential users cannot evaluate the security and effectiveness of the TON Blockchain (and the utility of the Gram token) until these software components have been developed, released, and their security and performance evaluated and established.
- The TON public documents describe a suite of services that will eventually be purchasable by Gram holders. Today, however, few if any of these services exist, based on my review, and the TON Blockchain is not yet mature enough to support them. Here, too, the value of the services purchasable by the Gram token cannot be evaluated until Telegram and/or others develop and release the various services, and take measures to demonstrate that they are secure and perform well. Appendix C summarizes the current state of applications: the applications that involve the Gram token are either non-existent or awaiting integration with the TON Blockchain.

5. OVERVIEW OF BLOCKCHAINS AND CRYPTOCURRENCIES

9. A *blockchain* is just a ledger: an append-only list of *entries*, where an entry is typically a payment or asset transfer. A blockchain must be *tamper-proof*: an entry, once present, cannot be altered or erased. A blockchain is *distributed*: its data is replicated among members of a community, or sometimes the public at large, ensuring that a copy is always accessible to every interested party, even if governments or other actors attempt to suppress it. The most common use of a blockchain is to manage a *cryptocurrency*, or *coin*, a unit of accounting and store of value. The ledger records each coin's current owner and each transfer of ownership, ensuring that coins cannot be lost or duplicated.

10. Multiple nodes called *validators* ensure that only legitimate transactions are recorded on the blockchain. It is assumed that some subset of the validators may become corrupted and behave dishonestly. To protect against such corruption, the validators typically vote on each successive group (or *block*) of transactions to include in the ledger. Each block is linked, using tamper-proof cryptographic techniques, to its chronological predecessor block, forming an indelible chain of blocks (hence “blockchain”). To evaluate a blockchain design, it is essential to focus on the ways in which validators are selected, how they vote, how they are rewarded, and how corrupt validators are detected and deterred.

11. A *smart contract*² is a script executed by the validators and recorded on the blockchain, that controls activities such as creation and resolution of escrows, selection of participants, and participant rewards and penalties. Smart contracts play a central role in the design of the TON blockchain.

² A “smart contract” is neither “smart” nor is it a legal contract.

12. To be credible, a blockchain system, which takes custody of clients' money or other financial assets, must meet a higher degree of scrutiny than ordinary software systems. Such a system must not only *be* secure and efficient, it must be *provably* secure and efficient, otherwise no one will trust it enough to use it.

13. Employing virtuoso programmers, however skilled, is no guarantee of security. For example, in 2016, a team of celebrity programmers (including some of the founders of the popular Ethereum blockchain) established a blockchain-based hedge fund called the Decentralized Autonomous Organization, or DAO³. In one DAO smart contract, however, two apparently-unrelated steps were taken in the wrong order, opening a security vulnerability unnoticed by the DAO's implementers, who, as founders of the Ethereum Blockchain, the world's second largest blockchain, were ostensibly among the world's most experienced blockchain programmers at the time. Unknown parties almost immediately stole about \$55 million worth of cryptocurrency, a crime unsolved to this day⁴. Security is a dull and plodding business: a program that is 99.9% secure, however elegantly structured, is not secure at all.

14. Attacks may come from within, from corrupt insiders, or from without, from hostile consumers or third parties. Attackers may be well-funded: widely-used blockchains such as those proposed for Facebook or Telegram users may be the target of nation-state actors who attempt disruption for strategic reasons. Such attackers are even harder to protect against than attackers motivated only by greed.

15. Attacks on a blockchain can take many forms. Most obviously, an attack may attempt to steal assets through unauthorized transfers to the attacker, or to "burn" assets by making

³ [https://en.wikipedia.org/wiki/The_DAO_\(organization\)](https://en.wikipedia.org/wiki/The_DAO_(organization))

⁴ <https://www.bloomberg.com/features/2017-the-ether-thief/>

them inaccessible to their owners. A *denial of service* attack seeks to slow or prevent legitimate use of the blockchain. For example, an attacker might flood the ledger with many very low-value “dust” transactions, displacing legitimate transactions, or the attacker might corrupt just enough validators to prevent timely progress. In a *censorship* attack, specific parties may be targeted for disruption. For example, one hedge fund might “front-run” another’s stock orders to drive up the stock price, or one nation-state may impede a second nation-state’s transactions with the intention of disrupting the latter’s economy.

16. In the face of such a variety of possible attacks, how can a reasonable investor or consumer tell whether a blockchain design is secure? There are three practices commonly accepted in the industry.

- *Peer-reviewed publication*: the gold standard for establishing security is a publication at a peer-reviewed conference. For example, see publications on core blockchain algorithms for Facebook⁵ and Algorand^{6,7}.

⁵ Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness. In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC '19). ACM, New York, NY, USA, 347-356.

⁶ Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles (SOSP '17). ACM, New York, NY, USA, 51-68.

⁷ D. Leung, A. Suhl, Y. Gilad, N. Zeldovich. Vault: Fast Bootstrapping for the Algorand Cryptocurrency. In Network and Distributed System Security Symposium, 2019.

- *Self-published “white papers”*: organizations often self-publish documents that include mathematical analyses of security threats along with proofs of correctness. Examples include Algorand⁸, Facebook^{9,10}, and Stellar¹¹.
- *Third-party auditing*: An organization may engage an independent third party to audit and publicly certify its code.

17. Despite the complexity of the threats, and the value of the assets to be entrusted to its blockchain’s custody, Telegram, to my knowledge, has yet to take any of these measures, hence substantial work remains to be done to establish that the TON Blockchain is, or will be in the foreseeable future, secure.

6. TECHNICAL OVERVIEW OF TON BLOCKCHAIN

18. The TON Blockchain collects multiple prior blockchain design ideas and combines them into one complex, unified structure. Here I review its most important aspects, based on my review of the TON Blockchain public documents and public source code release.

A. TYPES OF NODES

19. *Validators* are the nodes (computers) that decide which transactions are allowed onto the blockchains (*i.e.*, which entries are included in the ledger). The validators form the very heart of the blockchain, since they decide which transactions are legitimate.

- For the TON Blockchain validators are to be elected, via a system-wide smart contract, about once a month. A node can propose itself as a validator, pledging a *stake* (a sum of Gram tokens) to ensure its good behavior. If the

⁸ <https://www.algorand.com/resources/white-papers/>

⁹ <https://developers.libra.org/docs/state-machine-replication-paper>

¹⁰ <https://developers.libra.org/docs/move-paper>

¹¹ <https://www.stellar.org/papers/stellar-consensus-protocol.pdf>

validator is inactive, or is detected to have behaved incorrectly, its stake can be reduced or confiscated entirely.

- A node that wishes to become a validator on the TON Blockchain must provide a suitable hardware environment. The minimum configuration recommended in the TON documentation¹² could be rented from Google Cloud Services for between \$500 and \$600 a month. Each would-be validator must also put up a stake of at least 100,001 Grams¹³. This sum is at risk if the validator goes off-line for too long (even if caused by a denial-of-service attack) or if the validator produces an incorrect result (even if caused by a system bug or honest mistake). If there is competition, the higher stake is preferred, so a winning validator may have to pledge a substantially higher stake than the minimum.
- Running a validator node on the TON Blockchain is not for the casual TON enthusiast. Working with an Integra Data Analyst, we were able to set up and run a “Lite” client, and then a full node, both prerequisites for running a validator node. Over and beyond the expense and risk described above, running a validator node is comparable to running a small data center: it will require substantial technical expertise configuring and maintaining servers, software, and network interfaces, and a long-term commitment to do so.

¹² “We recommend a dual-processor server with at least eight cores in each processor, at least 256 MiB RAM, at least 8 TB of conventional HDD storage and at least 512 GB of faster SSD storage, with 1 Gbit/s network (and Internet) connectivity to reliably accomodate [*sic*] peak loads.” In file doc/Validator-HOWTO in the release code.

¹³ About \$400,000 at the last price cited in a website purporting to have engaged in the sale of Grams to the public: <https://www.liquid.com/gram/>

- Validators are to be rewarded for their work through various fees and newly-minted Grams, in proportion to their stakes.
- Validators conduct a *Byzantine-Fault-Tolerant consensus* (BFT consensus) protocol to collectively decide which block to append next to the blockchain. Such protocols typically work correctly as long as no more than one-third of the validators (weighted by stake) are corrupted. The BFT consensus protocol lies at the heart of the TON Blockchain, and its security and performance are critical to the entire system.

20. A *nominator* node invests in validators, lending money, and sharing the validator's gains or losses, all governed by system-supplied smart contracts.

21. A *collator* node combines transactions into blocks, and suggests the blocks to validators, saving them work, and earning a fee, all governed by system-supplied smart contracts.

22. A *fisherman* is a node that checks the work of validators. If a fisherman can prove that a validator approved an illegal block, the validator's stake is reduced, and the fisherman receives a reward, all governed by system-supplied smart contracts.

B. SCALABILITY MECHANISMS

23. To improve scalability (ensuring that a blockchain performs well as the number of participants increases) activities on the TON Blockchain are split among a *masterchain* and a number of *workchains*. Most of the activity takes place on the workchains, and workchain states are periodically checkpointed to the main chain. In principle, different workchains can have different rules, although the current public release supports only one master and one workchain.

24. To further improve scalability, a workchain on the TON Blockchain can be split into multiple *shardchains*, where each shardchain is managed by only a subset of validators. Shardchains can be split and merged dynamically as the validator load changes.

25. A *vertical blockchain* is TON terminology for a way to fix a block incorrectly approved by corrupt validators. Once an invalid block has been detected, it is effectively replaced by another block, and changes are propagated as needed to dependent blocks later in the chain.

C. COMMUNICATION PROTOCOLS

26. Sometimes transactions on different shardchains need to communicate. *Hypercube routing* sends messages along a path designed to avoid congestion, where intermediate nodes are required to forward the message or lose a portion of their stake. *Instant hypercube routing* is a faster “best-effort” variation in which intermediate nodes are not required to participate.

7. EVALUATION OF TON BLOCKCHAIN CODE

27. I inspected the code in the public release with the intent of determining which essential system components exist, and which have yet to be fully implemented. I found there are several critical components that are either incomplete or missing.

28. The public documents do not disclose the BFT consensus protocol executed by the validators, and that protocol is difficult to reconstruct from the validator code¹⁴ in the public release. Moreover, it is impossible to tell whether the code represents the final version of the protocol. As discussed above, validators conduct a BFT consensus protocol to collectively decide which block to append next to the blockchain. The BFT consensus protocol lies at the heart of the TON Blockchain. Without a clear exposition of the BFT consensus protocol, along with a

¹⁴ Primarily in the `ton/validator`, `ton/validator-engine`, `ton/validator-session`, and `/crypto/smartcont` directories of the public source code.

thorough security and a performance analysis, I cannot consider the TON Blockchain close to deployment.

29. The public documents assert that a smart contract controls the creation of new workchains. I was unable to find any such contract in the public release. While it is natural that the “testnet” release would not permit such creation, the workchain-creation smart contract is a potential point of attack, and I cannot be convinced of the TON Blockchain’s security without a careful analysis of its code.

30. The public documents assert that a smart contract controls how absent or incorrect validators are punished. I was unable to find any such contract in the public release. The validator punishment smart contract is an inviting attack route, and I cannot be convinced of the TON Blockchain’s security without a careful analysis of its code.

31. Additionally, if a “fisherman” node detects that a validator approved an incorrect block, the fisherman presents a proof to the mainchain that a block approved by the validator is incorrect, resulting in punishment of the guilty validator (or former validator), as well as some form of reward for the fisherman, and a potentially cascading "vertical blockchain" adjustment to compensate for the error. I was unable to find code for rewarding the fishermen function. I cannot be convinced of the TON Blockchain’s security without a careful analysis of how correct accusations are rewarded and false accusations deterred, since there is a potential for denial-of-service attacks.

32. When a corrupted or incorrect block is detected, the public documents assert that the “vertical blockchain” mechanism is used to fix the error, creating a new block to shadow the old block, and potentially triggering cascading adjustments in later blocks. I was unable to find code for this functionality. I cannot be convinced of the TON Blockchain's security without a

careful analysis of the cascading error-recovery code, since complex algorithms are required to unwind the error correctly and efficiently.

33. As reported earlier, based on work conducted at my direction by an Integra Data Analyst, it appears that the current testnet version of the TON Blockchain has many fewer participating nodes and much lighter activity than the TON Blockchain will eventually have when it is fully launched. For example, only 36 validators were detected, while the TON documentation projects as many as 1000 validators when the system is fully running. Hence, it is impossible to predict from observing the testnet release whether the TON Blockchain will meet its performance and scalability goals. Given the difference in scale between the testnet version and the final release version, I believe substantial further engineering work will be required to “tune” the Ton Blockchain before it can reach the size and speed described in the public documents.

8. CONCLUSIONS

34. The public code release is missing a number of components critical to the TON Blockchain’s functionality, including the core BFT consensus protocol, and the mechanisms by which validators are selected, rewarded, and punished for bad behavior. A substantial amount of additional software development remains to be done before the TON Blockchain can operate as envisioned. This software includes performance and security-critical modules.

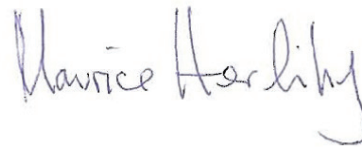
35. Neither the public documents, nor the structure of the publicly released code makes any attempt to analyze or prove the security of the system architecture, or its embodiment in code. I am not aware of any peer-reviewed papers analyzing the TON Blockchain. Telegram has published several white papers describing the proposed internal structure of their system, but none addresses security. No third-party code audits have been disclosed. A lack of security analysis does not necessarily imply the system is insecure, but no prudent investor or consumer would trust

their assets to such a system without a thorough security analysis and audit. Substantial work in this area remains to be done.

36. There is as yet no experimental evidence that I am aware of that the system is scalable to the degree envisaged by the founders. As reported earlier, the current TON Blockchain is running at a very small fraction of its eventual capacity, so it is impossible to draw reliable estimates of how well the TON Blockchain will perform when it is eventually deployed. A substantial amount of experimental measurements need to be performed, and reported in a clear and reproducible way, before I would be confident that the TON Blockchain is secure, scalable, and ready for release.

37. The collection of services surrounding the TON Blockchain and Gram token is largely unrealized. As summarized in Appendix C, of the applications listed by the Defendants' response, those that involve the Gram token are either undeveloped, or still awaiting integration with the TON Blockchain and the Gram token. A substantial amount of development and testing for these services must await the further development and testing of the core TON Blockchain itself.

38. I reserve the right to modify my analysis and conclusions based on the receipt of new information.

A handwritten signature in blue ink that reads "Maurice Herlihy". The signature is written in a cursive, slightly slanted style.

Maurice Herlihy

Appendix A

Curriculum Vitae of Maurice Herlihy

Maurice Peter Herlihy

Box 1910, Computer Science Dept

Brown University

Providence, RI 02912

<http://www.cs.brown.edu/people/mph/home.html>

(401) 863-7646 (voice)

(401) 863-7657 (fax)

herlihy@cs.brown.edu

Research Interests

Practical and theoretical aspects of concurrent and distributed systems.

Positions Brown University

Since 1994

An Wang Professor of Computer Science (since 2016), Professor (1998-2016), and Associate Professor (1994-1998).

1989 - 1994

Digital Equipment Corporation

Cambridge MA

Researcher, Cambridge Research Laboratory. Promoted to Consulting Engineer 1994.

1984 - 1989

Carnegie Mellon University

Pittsburgh PA

Assistant Professor, Computer Science Department.

October 2010 - August 2011

Technion

Haifa, Israel

Visiting (on sabbatical).

September 2004 - August 2005

Microsoft Research

Cambridge, UK

On leave, visiting researcher.

Honors and Awards

2017

Research Innovation Award, Brown University.

2016

Appointed to An Wang Chair of Computer Science. Brown University.

2015

Elected to the *American Academy of Arts and Sciences*.

SIGOPS Hall of Fame awarded to “Transactional Memory: architectural support for lock-free data structures.” In *Proceedings of the 1993 International Symposium on Computer Architecture*, May 1993, San Diego, CA.

2014

Elected fellow of the *National Academy of Inventors*.

2013

Elected to the *National Academy of Engineering*, “for concurrent computing techniques for linearizability, non-blocking data structures, and transactional memory”.

2013

the IEEE Computer Society’s *W. Wallace McDowell Award*, “for fundamental contributions to the theory and practice of multi-processor computation”.

2012

Dijkstra Prize in Distributed Computing. Awarded for “Transactional Memory: architectural support for lock-free data structures.” In *Proceedings of the 1993 International Symposium on Computer Architecture*, May 1993, San Diego, CA.

2010

Fulbright Distinguished Chair in the Natural Sciences and Engineering Lecturing Fellowship.

2008

International Symposium on Computer Architecture *Influential Paper Prize* awarded to *Transactional Memory* by Maurice Herlihy and J.E.B. Moss, International Symposium on Computer Architecture, 1993.

2005

Named *fellow of the Association for Computing Machinery* “for contributions to distributed and parallel systems.”

2004

Gödel Prize for outstanding journal articles in theoretical Computer Science.

2003

Dijkstra Prize in Distributed Computing. Awarded for “Wait-free synchronization”, *ACM Transactions on Programming Languages and Systems*, 13(1):124–149, January 1991.

Education

June 1984

Massachusetts Institute of Technology.
Ph.D. in Computer Science.

Cambridge MA

June 1980

Massachusetts Institute of Technology.
M.S. in Computer Science.

Cambridge MA

June 1975

Harvard University
A.B. in Mathematics.

Cambridge MA

Textbooks

Distributed Computing Through Combinatorial Topology, Morgan Kaufman 2013. ISBN 9780124045781.

The Art of Multiprocessor Programming by Maurice Herlihy and Nir Shavit. Morgan Kaufmann 2008. ISBN 0123705916.

Journal Publications (last 10 years)

M. Herlihy Blockchains from a distributed computing perspective. *Commun. ACM* 62(2): 78-85 (2019)

Maurice Herlihy, Liuba Shrira, Barbara Liskov: Cross-chain Deals and Adversarial Commerce. *PVLDB* 13(2): 100-113 (2019)

Hagit Attiya, Armando Castañeda, Maurice Herlihy, Ami Paz: Bounds on the Step and Namespace Complexity of Renaming. *SIAM J. Comput.* 48(1): 1-32 (2019)

Costas Busch, Maurice Herlihy, Miroslav Popovic, Gokarna Sharma: Time-communication impossibility results for distributed transactional memory. *Distributed Computing* 31(6): 471-487 (2018)

Dimitra Papagiannopoulou, Andrea Marongiu, Tali Moreshet, Luca Benini, Maurice Herlihy, R. Iris Bahar: Hardware Transactional Memory Exploration in Coherence-Free Many-Core Architectures. *International Journal of Parallel Programming* 46(6): 1304-1328 (2018)

Maurice Herlihy, Sergio Rajsbaum, Michel Raynal, Julien Stainer: From wait-free to arbitrary concurrent solo executions in colorless distributed computing. *Theor. Comput. Sci.* 683: 1-21 (2017)

Maurice Herlihy, Zhiyu Liu: Well-Structured Futures and Cache Locality. *ACM Transactions on Parallel Computing* 2(4): 22 (2016)

Hammurabi Mendes, Maurice Herlihy, Nitin H. Vaidya, Vijay K. Garg: Multi-dimensional agreement in Byzantine systems. *Distributed Computing* 28(6): 423-441 (2015)

Dimitra Papagiannopoulou, Giuseppe Capodanno, Tali Moreshet, Maurice Herlihy, R. Iris Bahar: Energy-Efficient and High-Performance Lock Speculation Hardware for Embedded Multicore Systems. *ACM Trans. Embedded Comput. Syst.* 14(3): 51 (2015)

Armando Castañeda, Maurice Herlihy, Sergio Rajsbaum: An Equivariance Theorem with Applications to Renaming. *Algorithmica* 70(2): 171-194 (2014)

Maurice Herlihy, Sergio Rajsbaum, Michel Raynal: Power and limits of distributed computing shared memory models. *Theor. Comput. Sci.* 509: 3-24 (2013) Maurice Herlihy, Sergio Rajsbaum: The topology of distributed adversaries. *Distributed Computing* 26(3): 173-192 (2013)

Maurice Herlihy, Sergio Rajsbaum, Michel Raynal: Power and limits of distributed computing shared memory models. *Theor. Comput. Sci.* 509: 3-24 (2013)

Flavio Paiva Junqueira, Keith Marzullo, Maurice Herlihy, Lucia Draque Penso: Threshold protocols in survivor set systems. *Distributed Computing* 23(2): 135-149 (2010)

Cesare Ferri, Samantha Wood, Tali Moreshet, R. Iris Bahar, Maurice Herlihy: Embedded-TM: Energy and complexity-effective hardware transactional memory for embedded multicore systems. *J. Parallel Distrib. Comput.* 70(10): 1042-1052 (2010)

Conference Publications (last 10 years)

Archita Agarwal, Maurice Herlihy, Seny Kamara, Tarik Moataz: Encrypted Databases for Differential Privacy. *PoPETs* 2019(3): 170-190 (2019)

Attacking memory-hard script with near-data-processing. *MEMSYS* 2019: 33-37 Samuel Irving, Sui Chen, Lu Peng, Costas Busch, Maurice Herlihy, Christopher J. Michael:

CUDA-DTM: Distributed Transactional Memory for GPU Clusters. *NETYS* 2019: 183-199 [c161] Jiwon Choe, Amy Huang, Tali Moreshet, Maurice Herlihy, R. Iris Bahar:

Concurrent Data Structures with Near-Data-Processing: an Architecture-Aware Implementation. *SPAA* 2019: 297-308

Vikram Saraph, Maurice Herlihy: An Empirical Study of Speculative Concurrency in Ethereum Smart Contracts. CoRR abs/1901.01376 (2019)

Dave Dice, Maurice Herlihy, Alex Kogan: Improving Parallelism in Hardware Transactional Memory. TACO 15(1): 9:1-9:24 (2018)

Victor Cacciari Miraldo, Harold Carr, Alex Kogan, Mark Moir, Maurice Herlihy: Authenticated modular maps in Haskell.

Maurice Herlihy: Atomic Cross-Chain Swaps. PODC 2018: 245-254

Michal Friedman, Maurice Herlihy, Virendra J. Marathe, Erez Petrank: A persistent lock-free queue for non-volatile memory. PPOPP 2018: 28-40

Shishir Rai, Gokarna Sharma, Costas Busch, Maurice Herlihy: Load Balanced Distributed Directories. SSS 2018: 221-238

Rida A. Bazzi, Maurice Herlihy: Clairvoyant State Machine Replications. SSS 2018: 254-268

Thomas D. Dickerson, Paul Gazzillo, Maurice Herlihy, Eric Koskinen: Adding Concurrency to Smart Contracts. PODC 2017: 303-312.

Costas Busch, Maurice Herlihy, Miroslav Popovic, Gokarna Sharma: Fast Scheduling in Distributed Transactional Memory. SPAA 2017: 173-182

Zhiyu Liu, Irina Calciu, Maurice Herlihy, Onur Mutlu: Concurrent Data Structures for Near-Memory Computing. SPAA 2017: 235-245

Hammurabi Mendes, Maurice Herlihy: Tight Bounds for Connectivity and Set Agreement in Byzantine Synchronous Systems. DISC 2017: 35:1-35:16

Dimitra Papagiannopoulou, Andrea Marongiu, Tali Moreshet, Maurice Herlihy, and R. Iris Bahar. 2017. Edge-TM: Exploiting Transactional Memory for Error Tolerance and Energy Efficiency. ACM Trans. Embed. Comput. Syst. 16, 5s, Article 153 (September 2017), 18 pages. DOI: <https://doi.org/10.1145/3126556>

Thomas Carle, Dimitra Papagiannopoulou, Tali Moreshet, Andrea Marongiu, Maurice Herlihy, R. Iris Bahar: Thrifty-malloc: A HW/SW codesign for the dynamic management of hardware transactional memory in embedded multicore systems. CASES 2016: 20:1-20:10

Dave Dice, Maurice Herlihy, Alex Kogan: Fast non-intrusive memory reclamation for highly-concurrent data structures. ISMM 2016: 36-45

Maurice Herlihy, Mark Moir: Blockchains and the Logic of Accountability. LICS 2016: 27-30

Oana Balmau, Rachid Guerraoui, Maurice Herlihy, Igor Zablotchi: Fast and Robust Memory Reclamation for Concurrent Data Structures. SPAA 2016: 349-359

Vikram Saraph, Maurice Herlihy, Eli Gafni: Asynchronous Computability Theorems for t-Resilient Systems. DISC 2016: 428-441

Shahar Timnat, Maurice Herlihy, Erez Petrank: A Practical Transactional Memory Interface. Euro-Par 2015: 387-401

Dimitra Papagiannopoulou, Andrea Marongiu, Tali Moreshet, Luca Benini, Maurice Herlihy, R. Iris Bahar: Playing with Fire: Transactional Memory Revisited for Error-Resilient and Energy-Efficient MPSoC Execution. ACM Great Lakes Symposium on VLSI 2015: 9-14

Costas Busch, Maurice Herlihy, Miroslav Popovic, Gokarna Sharma: Impossibility Results for Distributed Transactional Memory. PODC 2015: 207-215

Laurent Réveillère, Tim Harris, Maurice Herlihy: Proceedings of the Tenth European Conference on Computer Systems, EuroSys 2015, Bordeaux, France, April 21-24, 2015. ACM 2015, ISBN 978-1-4503-3238-5

Maurice Herlihy, Vikram Saraph: The Relative Power of Composite Loop Agreement Tasks. OPODIS 2015.

James A. Jablin, Thomas B. Jablin, Onur Mutlu, Maurice Herlihy: Warp-aware trace scheduling for GPUs. PACT 2014: 163-174

Irina Calciu, Justin Gottschlich, Tatiana Shpeisman, Gilles Pokam, Maurice Herlihy: Invyswell: a hybrid transactional memory for haswell's restricted transactional memory. PACT 2014: 187-200

Maurice Herlihy, Eric Koskinen: Composable Transactional Objects: A Position Paper. ESOP 2014: 1-7

Dan Alistarh, Patrick Eugster, Maurice Herlihy, Alexander Matveev, Nir Shavit: StackTrack: an automated transactional approach to concurrent memory reclamation. EuroSys 2014: 25

Eli Gafni, Maurice Herlihy: Sporadic Solutions to Zero-One Exclusion Tasks. ICALP (1) 2014: 1-10

Maurice Herlihy, Sergio Rajsbaum, Michel Raynal, Julien Stainer: Computing in the Presence of Concurrent Solo Executions. LATIN 2014: 214-225

Alex Kogan, Maurice Herlihy: The future(s) of shared data structures. PODC 2014: 30-39

Maurice Herlihy, Zhiyu Liu: Well-structured futures and cache locality. PPOPP 2014: 155-166

Dimitra Papagiannopoulou, Tali Moreshet, Andrea Marongiu, Luca Benini, Maurice Herlihy, R. Iris Bahar: Speculative synchronization for coherence-free embedded NUMA architectures. ICSAMOS 2014: 99-106 (*best paper award*).

Maurice Herlihy: Fun with hardware transactional memory. SIGMOD Conference 2014: 575

Hammurabi Mendes, Christine Tasson, Maurice Herlihy: Distributed computability in Byzantine asynchronous systems. STOC 2014: 704-713

Zhiyu Liu, Maurice Herlihy: Approximate Local Sums and Their Applications in Radio Networks. DISC 2014: 243-257

Irina Calciu, Hammurabi Mendes, Maurice Herlihy: The Adaptive Priority Queue with Elimination and Combining. DISC 2014: 406-420

Dimitra Papagiannopoulou, R. Iris Bahar, Tali Moreshet, Maurice Herlihy, Andrea Marongiu, Luca Benini: Transparent and energy-efficient speculation on NUMA architectures for embedded MPSoCs. MES 2013: 58-61

Hagit Attiya, Armando Castañeda, Maurice Herlihy, Ami Paz: Upper bound on the complexity of solving hard renaming. PODC 2013: 190-199

Hammurabi Mendes, Maurice Herlihy: Multidimensional approximate agreement in Byzantine asynchronous systems. STOC 2013: 391-400

Maurice Herlihy, Sergio Rajsbaum: Simulations and reductions for colorless tasks. PODC 2012: 253-260.

Armando Castañeda, Maurice Herlihy, Sergio Rajsbaum: An Equivariance Theorem with Applications to Renaming. LATIN 2012: 133-144.

Justin Emile Gottschlich, Maurice Herlihy, Gilles Pokam, Jeremy G. Siek: Visualizing transactional memory. PACT 2012: 159-170.

Cesare Ferri, Andrea Marongiu, Benjamin Lipton, R. Iris Bahar, Tali Moreshet, Luca Benini, Maurice Herlihy: SoC-TM: integrated HW/SW support for transactional memory programming on embedded MPSoCs. CODES+ISSS 2011: 39-48

Maurice Herlihy, Nir Shavit: On the Nature of Progress. OPODIS 2011: 313-328

Maurice Herlihy, Yoram Moses, Mark R. Tuttle: Transforming worst-case optimal solutions for simultaneous tasks into all-case optimal solutions. PODC 2011: 231-238

Aleksandar Dragojevic, Maurice Herlihy, Yossi Lev, Mark Moir: On the power of hardware transactional memory to simplify memory management. PODC 2011: 99-108

Armando Castañeda, Maurice Herlihy, Sergio Rajsbaum: An Equivariance Theorem with Applications to Renaming (Preliminary Version) CoRR abs/1102.4946: (2011)

Maurice Herlihy, Sergio Rajsbaum: Concurrent Computing and Shellable Complexes. DISC 2010: 109-123

Maurice Herlihy, Sergio Rajsbaum: The topology of shared-memory adversaries. PODC 2010: 105-113

Cesare Ferri, Samantha Wood, Tali Moreshet, R. Iris Bahar, Maurice Herlihy: Energy and Throughput Efficient Transactional Memory for Embedded Multicore Systems. HiPEAC 2010.

Eric Koskinen, Matthew Parkinson, Maurice Herlihy: Coarse-grained transactions. POPL 2010: 19-30.

Other Publications

Over 50 patents granted.

Appendix B

Documents Relied Upon

- <https://test.ton.org/ton.pdf>
- <https://test.ton.org/tvm.pdf>
- <https://test.ton.org/fiftbase.pdf>
- <https://test.ton.org/smc-guidelines.txt>
- <https://test.ton.org/README.txt>
- <https://test.ton.org/HOWTO.txt>
- <https://test.ton.org/FullNode-HOWTO.txt>
- [Complaint in *SEC v. Telegram Group Inc. and TON Issuer Inc.*](#)
- [Purchase Agreement TG-001-00000014-53](#)
- [Purchase Agreement TG-003-00000223](#)
- <https://test.ton.org/Validator-HOWTO.txt>
- [Telegram Objections and Responses to SEC Interrogatories](#)

Appendix C

Spreadsheet of Potential TON Applications

	B	C	D	E
	Name	Developer	Notes Based on Third Party Description	Type
2	SOL2TVM compiler	TON Labs	Tool to ensure contract compatibility between Ethereum platform and TON Virtual Machine	Compiler
3				
4				
5				
6				
7	LLVM compiler	TON Labs	LLVM-based compiler designed to convert sources from multiple high-level languages into its IR and then into TVM bytecode	Compiler
8				
9	TON Labs Local Node	TON Labs	TON Labs proprietary implementation of TON Node	Compiler
10				
11	TON Labs SDK	TON Labs	CLI tool for streamlined usage https://ton.dev/node-se	Compiler
12	TON Labs Toolchain	TON Labs	TON Labs Compiler kit with the latest versions of TON Labs LLVM and Sol2TVM compilers	Compiler
13				
14	Ton Labs Node SE	TON Labs	Provides a set of developer tools to develop and compile smart contracts in Solidity, C and C++; run, deploy and test contracts	Compiler
15	Button Wallet	Button	Wallet that supports BTC, ETH, LTC, BCH, ETC, WAVES, Stellar Lumens (XLM) and ERC-20 tokens; will facilitate exchange of Grams	Wallet
16				
17	Mercuryo Pay	Mercuryo	Allows customers to pay with BTC & ETH; will process Grams once TON Blockchain launches	Payment processor
18	AdGram	AdGram	Advertising platform that allows advertisers to create advertising campaigns and channel owners to monetize their audience	App/platform
19	BeProducers	BeProducers	Kickstarter for film production	App/platform
20	TON dApps Marketplace	CryptoBazar	Lists pre-selected projects to be launched on TON Blockchain	Developer
21	Drimsim SIM	Drimsim Global	Universal SIM and mobile expense card	App/platform
22				
23				
24	Denim	Denim	Dating service app	App/platform
25	Unovis	Unovis Forum	Marketplace for art using TON Blockchain	App/platform
26	DareApp	Eristica	Mobile video platform	App/platform
27	Posh.space	Posh.space	Digital fashion store	App/platform
28				
29	Pregnancy Tracker	Mobile Dimension LLC	Pregnancy related app	App/platform
30				
31	U-Robot	u-robot	Chat-bot app; customized online store or personal page	App/platform
32				
33	Incognito	Incognito	Mobile app to solicit anonymous feedback	App/platform
34	Kelvyn	Kelvyn	VPN operating on top of TON decentralized network	VPN
35				
36	EzDapps	Apla	Dapps Accelerator	Developer
37	Spatium	Spatium	Wallet with enterprise-level security	Wallet
38				
39	Viewst	Viewst	Browser-based editor of advertisement graphics and templates	App/platform
40	Worldwide Hackathon	Optimal.one	Worldwide Hackathon for the promotion of TON Blockchain projects	Hackathon
41	Parlar	Parachute	Allows users to send cryptocurrency tips on Telegram	Wallet
42	TON-based real-time advertising platform	Appreciate	Proposed real-time advertising platform on top of TON Blockchain	Advertising - However, no indication of actual app
43	copperbits/TON	Copperbits (on GitHub)	R&D group focused on TON Blockchain on GitHub	Informative
44				
45	TON.Broxus	Finex Future	Java wrapper for TON	API
46	Atomic TON Wallet	Atomic	Universal cryptocurrency wallet	Wallet
47	TON tokens	Emelyanenko K	Simple tokens for TON	Token
48	ton_client	formony	Python API client for TON	API
49	TON Watcher	TON Center	Will allow users to search for addresses or blocks on TON Blockchain	Block explorer
50	TON.shi Public API	TON.sh	HTTP-based experimental interface for developers; explorer to search for addresses or transactions on TON Blockchain	API

	F	G
	Gram used for in-app payment?	Existing product (e.g. website, application)?
2	No	
3		
4		
5		
6		
7	No	
8		
9	No	
10		
11	No	
12	No	
13		
14	No	
15	Converter	
16		
17	Converter	
18	Yes	Existing product; awaiting Gram integration
19	Yes	Existing product; awaiting Gram integration
20		
21	Yes	Existing product; awaiting Gram integration
22		
23		
24	Yes	Existing product; awaiting Gram integration
25	Yes	No
26	Yes	Existing product; awaiting Gram integration
27	Yes	No
28		
29	Yes	Existing product; awaiting Gram integration
30		
31	Yes	Existing product; awaiting Gram integration
32		
33	Yes	No
34	No	
35		
36	No	
37	Converter	
38		
39	Yes	Existing product; awaiting Gram integration
40	No	
41	Converter	
42		
43	No	
44		
45	No	
46	Converter	
47	No	
48	No	
49	No	
50	No	

	H
	Link(s)
2	https://www.coindesk.com/telegrams-blockchain-will-be-compatible-with-ethereum-tonlabs-says
3	https://cryptobriefing.com/telegram-ton-labs/
4	https://coingecko.com/news/report-telegrams-ton-blockchain-to-be-compatible-with-ethereum-dapps
5	https://docs.ton.dev/86757ecb2/p04a4ba
6	https://ton.dev/toolchain
7	https://docs.ton.dev/86757ecb2/p04a4ba
8	https://ton.dev/node-se
9	https://docs.ton.dev/86757ecb2/p04a4ba
10	https://docs.ton.dev/86757ecb2/p04a4ba
11	https://docs.ton.dev/86757ecb2/p04a4ba
12	https://ton.dev/node-se
13	https://docs.ton.dev/86757ecb2/p04a4ba
14	https://ton.dev/node-se
15	https://www.forbes.com/sites/billybambrough/2019/08/26/telegrams-300-million-users-could-soon-be-trading-bitcoin-and-cryptodespite-serious-security-warning/#1aa7fb523fe9
16	https://buttonwallet.com/
17	https://mercuryo.io/business/acquiring/
18	https://adgram.io/
19	https://beproducer.pro/news-and-analysis/d1f08a5e-262f-48d387be-020530e2317d
20	https://ton.cryptobazar.io/
21	https://m.facebook.com/334908950536095/posts/432905817403074?d=n&substory_index=0&sfs=mo
22	https://twitter.com/drimsimqlobal/status/1197175207649304577?s=21
23	https://blog.drimsim.com/drimsim-budet-prinimat-crypto-gram
24	https://dcentrzd.app/2019/11/17/denim/
25	https://dcentrzd.app/2019/10/23/unovis/
26	https://dcentrzd.app/2019/10/13/dareapp/
27	https://posh.space/
28	https://dcentrzd.app/2019/10/12/poshspace/
29	https://dcentrzd.app/2019/10/10/pregnancy-tracker/
30	https://pregnancytracker.app/
31	https://dcentrzd.app/2019/10/09/urobot/
32	http://u-robot.net
33	https://dcentrzd.app/2019/10/08/incognito/
34	https://dcentrzd.app/2019/10/07/kelvpn/
35	https://kelvpn.com/
36	https://dcentrzd.app/2019/10/06/ezapps/
37	https://dcentrzd.app/2019/10/05/spatium/
38	https://spatium.net/
39	https://viewst.com/
40	https://www.optimal.one/
41	https://beincrypto.com/cryptocurrency-tips-on-telegram-reach500000-milestone-in-just-a-year/
42	
43	https://github.com/copperbits/TON
44	https://t.me/ton_research
45	https://github.com/proxus/tonclient
46	https://atomicwallet.io/ton-wallet
47	https://github.com/EmelyanenkoK/TON_tokens
48	https://github.com/formony/ton_client
49	https://tonwatcher.com/
50	https://ton.sh/api